SYSTEM FOR WIRELESS PUSH AND PULL
BASED SERVICES

Background of the Invention

1. Technical Field

The invention relates to a proxy gateway for providing improved push and pull based services from a content provider on the Internet to a mobile user on a wireless network.

10

2. Description of Related Art

The Internet is a global network formed by the cooperative interconnection of computing networks. The Worldwide Web (WWW or Web) is a collection of files or "Web pages" of text, graphics and other media which are connected by hyperlinks to other Web pages which physically reside on the Internet. In a transaction on the WWW, a Web client typically requests information from a Web server. The requested information is transmitted from the Web server to the Web client over the Internet. Dramatically increasing expansion of Internet services using the WWW has led to increased Web traffic.

A conventional technique to reduce Web traffic and speed up Web access is to store copies of documents in a cache. U.S. Patent No. 5,873,100 describes an Internet browser which includes an embedded cache for user controlled document retention. The cache stores a plurality of documents. At least one of the documents stored in the cache is designated as a keep document. If the storage limit of the cache is exceeded, the cache deletes the oldest document not designated as a keep document.

Web servers and Web client use hypertext transfer protocol (HTTP) / 1.1 which includes cache control features. See R. Fielding et al., "Hypertext Transport Protocol HTTP / 1.1" Network Working Group RFC, May 1996, URL: ftp://ftp.isi.edu/in-notes/rfc2068.txt. The original Web server assigns expiration times to responses generated for Web client requests. An expiration judgment is performed in the cache when a cached entry is requested by a client. If the cached entry has not expired, the cache sends the entry to the client; otherwise, it sends a conditional request to the Web

server. A validation check is performed at the Web server to validate if the cached entry is still useable. If the cached entry is useable, the Web server sends a validator to the Web client; otherwise, it sends an updated response.

5   In certain systems, there exists very little local memory. In these systems, caching and prefetching is preferably performed at a proxy server intermediate between the Web server and the Web client. Prefetching is a technique in which additional items are fetched when a request is made for a particular item. U.S. Patent No. 5,925,100 describes a system having a dumb client environment in which a smart server determines when to send a prefetched object to the user. The prefetched objects are determined

10   based on an object based prefetch primitive present in the client's executing application.

Other conventional prefetch schemes are based on predicting at a given time the likelihood that a given document will be accessed in the near future. Prefetch schemes have been described in which a prediction module computes the access probability that a file will be requested in the near future. Each file whose access probability exceeds a

15   server's prefetch threshold is prefetched. See Z. Jiang and L. Kleinrock, "An Adaptive Network Prefetch Scheme," *IEEE J. on Selec. Areas in Common.*, vol. 16, no. 3, April 1998, pp. 358-368, and Z. Jiang and L. Kleinrock, "Web Prefetching in a Client Environment," *IEEE Personal Communications*, vol. 5, no. 5, Oct. 1998, pp. 25-34.

In addition, prefetch schemes have been described which are based on popularity

20   based prefetching. See E. P. Markatos, "Main Memory Caching of Web Documents," *Computer Networks and ISDN Systems*, vol. 28, issues 7-11, pp. 893-906, 1996. Main menu caching of frequently requested documents is performed on the Web server. Similarly, U.S. Patent No. 5,991,306 describes a pull based intelligent caching system for delivering data over the Internet. Content of frequently requested documents is

25   downloaded from the content provider and cached at a local service provider. The content is cached prior to a peak time when subscribers are likely to request the content. A pattern recognizer detects behavior patterns based on subscriber requests to determine which content the subscribers are most likely to request and when. When content is finally requested, the data is streamed continuously for rendering at the subscriber

30   computer.

2

Another prefetching scheme type is based on interactive prefetching in which prefetching is determined by the interaction between the client and the server. For example, an interactive prefetching scheme has been proposed in which the system gathers references by passing hypertext markup language (HTML) in the referenced page and collecting referenced pages with each request. See K. Chinen and S. Yamaguchi, "An Interactive Prefetching Proxy Server for Improvement of WWW Latency, " *INET'97*, Kuala Lumpur, Malaysia, 1997.

Wireless systems and mobile users are typically limited to small bandwidth and small memory. A wireless application protocol (WAP) has been developed to promote industry-wide specifications for technology useful in developing applications and services, such as Internet services, that operate over wireless communication networks, as described in Wireless Architecture Protocol Specification, Wireless Application Protocol Forum, Ltd., Version 30, April 1998 and WAP Push Architectural Overview, Version 08, November 1999. WAP framework defines pull technology as a transaction initiated by the client to pull information from a server. For example, the World Wide Web is an example of pull technology in which a user enters a universal resource locator (URL) which is sent to the server and the server sends a Web page to the user. WAP framework defines push technology as a transmission to the client without previous action by the client. Accordingly, the server pushes information to the client without an explicit request from the client. It is desirable to provide a system for wireless push and pull based Internet services which expeditiously allows users to gain access to desired Web information.

Summary of the Invention

The present invention relates to a method and system for providing Web content from pull and push based services running on Web content providers to mobile users. A proxy gateway connects the mobile users to the Web content providers. A prefetching module is used at the proxy gateway to optimize performance of the pull services by reducing average access latency. The prefetch module prioritizes pull content to be stored in a cache at the proxy gateway. The average access latency can be reduced by using at least one factor related to the frequency of access to the pull content, the update

cycle of the pull content determined by the Web content providers and the response delay for fetching pull content from the content provider to the proxy gateway. Pull content, such as documents, having the greatest average access latency are sorted and a predetermined number of the documents are prefetched into the cache. Push services are

5 optimized by iteratively estimating a state of each of the mobile users to determine relevant push content to be forward to the mobile user. The estimate of the state of each mobile user can be determined from tracking information of the mobile user and geo-location measurement and behavior observation data.

The invention will be more fully described by reference to the following

10 drawings.

Brief Description of the Drawings

Fig. 1 is a schematic diagram of a system for providing push and pull based services to mobile users.

15 Fig. 2 is a flow diagram of an implementation of a prefetch module used in a proxy gateway of the system.

Fig. 3 is an illustration of a caching model.

Fig. 4 is a schematic diagram for providing push services of the system.

Fig. 5 is a flow diagram of an implementation of the mobile state prediction.

20

Detailed Description

Reference will now be made in greater detail to a preferred embodiment of the invention, an example of which is illustrated in the accompanying drawings. Wherever possible, the same reference numerals will be used throughout the drawings and the

25 description to refer to the same or like parts.

Fig. 1 illustrates a schematic diagram of a system for providing push and pull based services to mobile users 10. Mobile users 12a-12n are connected by mobile network 11 to proxy gateway 13. For example, mobile network 11 is a wireless network. Web server 14 is connected by network 15 to proxy gateway 13. For example, network

30 15 can be a wired network, such as the Internet.

Each of mobile users 12a-12n can interact with pull service 16 and push service 20 running on content provider 14, such as a Web server. As an example of pull service 16, mobile user 12a generates a pull request 17 which is transmitted over network 11 to gateway 13. Pull request 17 is transmitted via proxy gateway 13 to pull service 16.

5 Mobile user 12a receives pull response 18 generated by pull service 16 at content provider 14 via proxy gateway 13. Pull response 18 includes pull content 19 stored or generated by content provider 14. As an example of push service 20, push message 21 is sent by push service 20 to mobile user 12b. Push message 21 includes push content 22 stored or generated at content provider 14. Push message 21 is sent by content provider

10 14 to proxy gateway 13. Push message 21 is forwarded by proxy gateway 13 to mobile user 12b.

Pull service 16 and push service 20 interact with cache 23, as described below. Pull response 18 generated by content provider 14 can be prefetched and cached in cache 23. Thereafter, when pull request 17 identifies a pull response 18 stored in cache 23, pull

15 response 18 stored in cache 23 can be forwarded to mobile user 12a, or any other mobile user 12b-12n, without contacting content provider 14. Prefetch module 24 reduces access latency of pull service 16 using document access log data 25 to determine prefetch control information 26 for prefetching a predetermined number of pull responses 18 into cache 23. Unprefetched pull response 18 is cached by using a conventional caching

20 mechanism, such as HTTP/1.1 caching. Mobile state estimation pedication module 27 uses mobile tracking data 28 and geo-location measurement and behavior observation data 29 to determine push control information 30a for controlling caching of push messages 21 in cache 23 and for determining timing for sending relevant push content to the mobile user 30b depending on the state of mobile user 12b. It will be appreciated that

25 multiple pull services 16 and push services 20 can be provided at content provider 14 or a plurality of content providers 14 and accessed by multiple mobile users 12a-12n. Pull content 19 and push content 22 can comprise, for example, documents including Hypertext Markup Language (HTML) files, Java files, and embedded data including images, video files and audio files.

30 Fig. 2 illustrates a flow diagram for performing prefetch module 24. Prefetch module 24 optimizes performance of pull service 16 by reducing average access latency

using at least one factor related to the frequency of access to pull content 19, the update cycle of pull content 19, and the response delay for fetching pull content 19 from Web server 14 to cache 23. Access latency is defined as the time between when pull request 17 is transmitted from a mobile user 12a-12n to the time when pull response 18 is

5   received at a mobile user 12a-12n. Web server 14 generates content header 31 which is forwarded as a header of pull response 18. For example, content header 31 can include fields as defined in HTTP/1.1 such as: Expires time, relating to the expiration time of pull content 19; Last-Modified time, relating to the time pull content 19 was last modified at Web server 14; and Content-Length, relating to the size of pull content 19. Proxy

10   gateway 13 receives content header 31 and stores fields of content header 31 in document access log data 25. Proxy gateway 13 also receives pull request 17 and stores fields related to pull request 17. Fields related to cache 23 are also stored in document access log data 25. For example, fields stored in document access log data 25 can include fields such as: request_time, related to the local time when proxy gateway 13 made pull request

15   17 to content provider 14; and response_time, related to the local time when cache 23 received pull response 18 from content provider 14.

In block 32, variables related to pull service 16 are extracted from document access log data 25. For example, the extraction block 35 can calculate the parameters associated with potential candidates that may be included into the list of the prefetched

20   documents. The document that has only one access record (i.e., no revisit) in the log or has not more than one update cycle during the statistic period is not treated as a potential candidate. Let R_n be the average rate of access to pull content 19. Pull content 19 is referred to as document n. R_n can be determined at proxy gateway 13 based on log data of cache 23 and content header 31 of pull responses 18 from content providers 14 and

25   stored in document access log data 25. Let s_n be the size of document n. For example, s_n can be determined from the Content-Length field forwarded by content provider 14 and stored in document access log data 25.

Let ΔT_n be the average response delay imposed by network 15 which can be measured by the time from when pull request 17 from a mobile user 12a-12n is

30   reformatted and forwarded by proxy gateway 13 to content provider 14 to when pull content 19 is fetched from or validated by content provider 14 to cache 23, i.e. which can

be measured by the difference between the logged request_time and response_time. $T_{s,n}$ is the average time delay imposed by network 11 and network 15, which is the time from when, pull request 17 is generated from mobile user 12a where pull request 17 is reformatted and is forwarded by proxy gateway 13 to content provider 14 and pull content 19 is fetched from or validated by content provider 14 as pull response 18 to proxy gateway 13 to when response 18 is received by mobile user 12a. $T_{c,n}$ is the average time delay imposed by network 11, which is the time between the generation of pull request 17 from mobile user 12a and the reception of pull response 18 by mobile user 12a when a valid copy of pull content 19, document n, is found in cache 23, including transmission time of pull response 18, roundtrip time from cache 23 to mobile user 12a and cache 23 processing time. Approximately, $\Delta T_n = T_{s,n} - T_{c,n}$. Let μ_n be the update cycle of pull content 19, document n, which is the average length of time between two successive expiration times or two successive modifications of pull content 19, document n.

The caching model of HTTP/1.1 is illustrated in Fig. 3 in which t_k represents the Last_Modified time and "○" represents the Expires time of pull content 19, document n generated by Web server 14. Cache 23 is accessed by a group of mobile users 12a-12n and N is the total number of documents n residing in various Web servers 14 with n = 1, ... N. Pull request 17a generated by one of mobile users 12a-12n and forwarded to cache 23 in cycle, μ_n, cannot be satisfied by cache 23 which is denoted by missing 1. Cache 23 fetches a copy of pull content 19 for pull request 17a from Web server 14. Consequent pull requests 17b and 17c generated by one of mobile users 12a-12n in cycle, μ_n, are satisfied by cache 23 which is denoted by hit 1 and hit 2. Thereafter, in a second cycle, μ_n, pull request 17d generated by one of mobile users 12a-12n and forwarded to cache 23 in cycle, μ_n, cannot be satisfied by cache 23 which is denoted by missing 2. Cache 23 fetches a fresh copy of pull content 19 for pull request 17d from Web server 14. Consequent pull requests 17e-g generated by one of mobile users 12a-12n in cycle, μ_n, are satisfied by cache 23 which are denoted respectively by hits 3-5. In a third cycle, μ_n, pull request 17h generated by one of mobile users 12a-12n and forwarded to cache 23 in cycle, μ_n, cannot be satisfied by cache 23 which is denoted by missing 3. Thereafter, pull request 17i generated by one of mobile users 12a-12n arrives at cache 23 between the

expiration time and end of cycle, μ_n. In this case, cache 23 validates pull content 19 at Web server 14, represented by validate 1, before using pull content 19 stored in cache 23. The distribution of interarrival time of pull requests 17a-17i to pull content 19 represented by document n is represented by $f_n(t)$ which can be an exponential distribution. Since Web server 14 typically specifies the expires time based on its schedule to the end of cycle, μ_n, the interval between the Expires time and the end of the cycle can have a stochastic or deterministic distribution.

In block 34 of Fig. 2, the access probability of access to document n, represented by γ_n is determined by:

$$\gamma_n = R_n / R \qquad\qquad (1)$$

wherein R is the total rate of access traffic on network 15 from gateway 13, which is the sum of R_n for n=1, 2, ..., N.

In block 36, the average hit rate for document n, represented by h_n, is determined by:

$$h_n = 1 - \frac{g_n}{R_n \mu_n}, \quad n=1, 2, ..., N, \qquad\qquad (2)$$

in which:

g_n is the probability that there is at least one request to document n during a given update cycle, μ_n, given by:

$$g_n = 1 - e^{-R_n \mu_n}, \quad n = 1, 2, ..., N, \qquad\qquad (3)$$

and

$R_n \mu_n$ is the expected number of accesses to document n in an update cycle of document n.

In block 38, the wired network access latency, represented by η_n, imposed by network 15 when the request is the first one for document n in the update cycle μ_n or Expires time for document n has been exceeded, as shown in Fig. 3, is computed from:

$$\eta_n = \gamma_n (1-h_n) \Delta T_n, \quad n = 1, ..., N, \qquad\qquad (4)$$

The values at η_n, n = 1, ..., N are sorted in descending order with document 1 having the greatest average latency imposed by network 15 labeled as η_1, and document N having the least average latency labeled as η_N and relabeled as: $\eta_1 \geq \eta_2 \geq \eta_3 \geq ... \geq \eta_N$.

5   In block 40, the total number of documents n to be prefetched to cache 23, represented by r, is determined by considering at least one factor. Examples of factors include: spare capacity of cache 23 that can be utilized by the prefetching after providing sufficient capacity for conventional caching, ΔC; spare transmission bandwidth, ΔB, of network 15; and desired improvement of hit probability, ΔH. The total number of

10   documents to be prefetched represented by r satisfies all of the following constraints:

The constraint of spare cache capacity, ΔC, is given by

$$\sum_{n=1}^{r} s_n (1-h_n) \leq \Delta C; \qquad (5)$$

where $\Delta C \approx C - \sum_{n=1}^{N} s_n h_n$, C is given capacity of cache 23 and $\sum_{n=1}^{N} s_n h_n$ is the capacity required for conventional caching such as described in the caching model of HTTP/1.1.

15   The constraint of spare transmission bandwidth, ΔB, on network 15 is given by:

$$\sum_{n=1}^{r} (1-g_n) \frac{s_n}{\mu_n} \leq \Delta B; \qquad (6)$$

where $\Delta B \approx B - \sum_{n-1}^{N} g_n \frac{s_n}{\mu_n}$, B is given bandwidth and $\sum_{n-1}^{N} g_n \frac{s_n}{\mu_n}$ is the bandwidth required for conventional caching.

The constraint of desired minimum improvement of hit probability, ΔH, is given by:

20

$$\sum_{n=1}^{r} \gamma_n (1-h_n) \geq \Delta H \qquad (7)$$

where $\Delta H \approx H - \sum_{n=1}^{N} \gamma_n h_n$, H is given hit probability, and $\sum_{n=1}^{N} \gamma_n h_n$ is the total average hit probability for conventional caching.

In block 41, the total number of documents to be prefetched to cache 23, r, that correspond to the r largest are selected and relabeled as $\eta_1, ..., \eta_r$. In block 42, the

25   documents determined in block 41 are prefetched into cache 23 at proxy gateway 13 as

soon as the documents expire at cache 23, which means the prefetched document n in

cache 23 is updated with average cycle, μ_n, as shown in block 43. Thereafter, blocks 32-

43 are repeated with a given period, such as several hours, several days, or several times

the expected update cycle, $\bar{\mu}$, is represented by:

$$\bar{\mu} = \sum_{h=1}^{N} \gamma_n \mu_n \qquad (8')$$

Accordingly, average latency for pull service 16 is derived as:

$$L = \sum_{n-1}^{N} \gamma_n \, [h_n T_{c,n} + (1-h_n)T_{s,n}] - \sum_{n-1}^{r} \eta_n \qquad (8)$$

In Eq. (8), the first term is the latency of a conventional cache scheme and is independent

of the prefetch scheme used and the second term $\sum_{n-1}^{r} \eta_n$ is the latency reduction as a result

of prefetch scheme of the present invention. The latency reduction is determined by the

number r and the selection of r prefetched documents. Thus, performance of blocks 32-

43 minimizes the average latency L.

Fig. 4 illustrates a schematic diagram for providing push services of the system.

Movement and behavior of the mobile user 50 is measured by geo-location measurement

and behavior observation block 29. An example of movement and behavior of the

mobile user data 50 is represented in Table 1 illustrating examples of different states a

mobile user 12a-12n can occupy.

Table 1 Mobile States

State	Description	Position and Behavior	Time	Speed	Direction	Mean dwell time in the state
State 0	Inactive (power off or out off location-dependent services).	--	--	--	--	d_0
State 1	Walking on a street.	X_1	t_1	$\approx 1 m/s$	along one direction	d_1
State 2	In a shopping mall.	X_2	open hours	≈ 0		d_2
State 3	Drive on a highway.	X_3	t_3	$\approx 30 m/s$	along one direction	d_3
State M-1						d_{MH}

State 0 is an inactive state in which mobile user 12a-12n can occupy when powering off its mobile terminal or is out of location-dependent services of wireless network 11. State 1 to State M-1 are active states which mobile user 12a-12n can occupy

5   while being actively involved with network 11 and interacting with pull service 16 and push service 20. Each state is determined by several parameters such as: position and behavior of mobile user 12a-12n, at time t represented by X_t, time of determining state, represented by t; speed of mobile user 12a-12n; direction of movement of mobile user 12a-12n; and mean dwell time in state m, represented by d_m wherein M is the total

10   number of distinct states defined for mobile user 12a-12n, and m=0, 1, ..., M-1. Y_t represents the results of geo-location position measurement and behavior observations of mobile user 12a-12n at time t. It is observed that Y_t is the observed or measured value of the position and behavior of mobile user 12a-12n and is in general different from X_t which is the true, but unknown, position and behavior because of geo-location and

15   estimation error. Geo-location position and behavior data 29 can be estimated with conventional methods as described in J.H. Reed, K.J. Krizman, B.D. Woerner and T.S. Rappaport, "An Overview of the Challenges and Progress in Meeting the E-911 Requirement for Location Service", IEEE Communications Magazine, Vol. 36, No. 4, April 1998, pp. 30-37, hereby incorporated by reference into this application.

20   S_t is the state of mobile user 12a-12n at time t wherein $S_t \in \{0, 1, ..., M-1\}$.

Mobile user 12a-12n can transit from one state to another. The transit mobility of one of mobile users 12a-12n, mobile user 12, can be represented by an M-state Markov chain with transition probability matrix (TPM):

$$P=[p_{mn}]_{M \times M} \qquad\qquad (9)$$

25   where the p_{mn} is the probability of transition from state m to state n, and m, n=0, 1, ..., M-1.

Tracking data 28 represents sequence data of mobile user 12 over time. Tracking data 28 includes Y_1^t and $\{\alpha_\tau(m), \tau=1, ..., t, m=1, ..., m-1\}$

S_1^t represents the state sequence of mobile user 12 from time 1 to t. X_1^t represents the

30   corresponding position and behavior sequence of mobile user 12 from time 1 to t.

Y_1^t represents the corresponding geo-location and observation sequence of mobile user 12 from time 1 to t. Mobile state prediction module 27 can generate push control in information 30 for controlling caching of push content 22 and for determining timing for sending push content 30b in cache 23 to mobile user 12 based on determined states.

5   Fig. 5 is a flow diagram of an implementation of mobile state prediction module 27. Mobile state prediction module 27 estimates the posteriori probability of the states of mobile user 12a-12n for a given geo-location and observation sequence, Y_1^t.

In block 60, the initial state of mobile user 12a-12n is defined upon registration of mobile user 12a-12n in system 10. Let $\alpha_t(m)$ represent a forward variable for state
10   sequence estimation which is a probability that the State at time t is m and the corresponding geo-location and observation sequence is Y_1^t.

The forward variables for state sequence estimation for mobile user 12 in the initial time are:

$$\alpha_0(0)=1, \qquad \alpha_0(m)=0, \qquad \text{for m=1, 2, ..., M-1.}$$

15   In block 62, a measured or observed value of a current geo-location position and behavior of mobile user 12, Y_t, is determined for determining geo-location measurement and behavior observation data 29.

In block 63, the probability $Pr\{Y_t \mid X\}$ that the geo-location measured result is Y_t when mobile user 12 position and behavior is X is predetermined by the geolocation and
20   observation error distribution.

In blocks 65 the values required for iteration are stored in a database which values can be mobile tracking data 28. In block 64, state sequence estimation variable $\alpha_t(m)$ for all m=1, 2 ... M-1 is determined, which is stored into the database for the next recursive computation.

25   The following analysis can be used for performing block 64. At time t, tracking data is represented by: $Y_1^t = (Y_1, Y_2, ..., Y_{t-1}, Y_t)$ where Y_t is the current measured data of the position and the behavior of mobile user 12 from the geo-location measurement and the collected information of proxy gateway 13. $\alpha_t(m)$ is computed for all m=1, 2, ..., M-1, by iterations from 1 through t from the following:

30    $$\alpha_t(m) = \sum_{m'=0}^{M-1} Pr\{S_{t-1} = m'; S_t = m; Y_1^t\}, \qquad (10)$$

$$= \sum_{m'=0}^{M-1} \text{Pr} \{S_{t-1} = m'; Y_1^{t-1}\} \, \text{Pr} \{S_t = m; Y_t | S_{t-1} = m'\}, \qquad (11)$$

$$= \sum_{m'=0}^{M-1} \alpha_{t-1} (m') \, p_{m'm} \sum_x \text{Pr} \{x | m\} \text{Pr} \{Y_t | x\} \qquad (12)$$

wherein $p_{m'm}$ is the state transition probability of mobile user 12, $\text{Pr}\{x | m\}$ is the probability that the mobile user locates at position and behavior as x when it is in state m at time t, for example the output probability of the Markov source, and $\text{Pr}\{Y_t | x\}$ is the probability that the geo-location measured result is Y_t when the mobile's position and behavior is x at time t.

In block 66, state z is determined by:

$$z = \arg \max_m \{\text{Pr}\{S_t = m | Y_1^t\} | m = 1,2,...,M-1\}, \qquad (13)$$

$$= \arg \max_m \{ \frac{\text{Pr}\{S_t = m; Y_1^t\}}{\text{Pr}\{Y_1^t\}} | m = 1,2,...,M-1\}, \qquad (14)$$

$$= \arg \max_m \{\alpha_t(m) | m = 1,2,...,M-1\}. \qquad (15)$$

Accordingly, blocks 60, 63, 64 and 66 determine mobile state prediction module 27.

In block 68, state z related push content 22 is pushed to mobile user 12a-12n as relevant push content to the mobile user 30b. In block 69, mobile user 12 may accept or reject push content 22 and this behavior is observed by block 62. The mobile user's current behavior and position is forwarded to block 62 and block 62-69 can be repeated.

It is to be understood that the above-described embodiments are illustrative of only a few of the many possible specific embodiments which can represent applications of the principles of the invention. Numerous and varied other arrangements can be readily devised in accordance with these principles by those skilled in the art without departing from the spirit and scope of the invention.